Packet Scheduling

Jon Turner
Computer Science & Engineering
Washington University

www.arl.wustl.edu/~jst
Motivation for Packet Scheduling

- Internet traffic subject to wide variations
  - no direct control over traffic from end systems
  - route changes cause large changes in traffic flows
  - links may be overloaded for long periods of time relative to link time scales (100s of ms vs. fractions of μs)

- Per flow with fair queueing provides traffic isolation
  - well-behaved flows get their fair share of congested link
    - so, minimal queuing delay
    - greedy flows’ packets get backed up or lost

- Packet scheduling key to good performance
  - most schedulers ensure fair share of link bandwidth
  - best schedulers can provide end-to-end delay guarantees
  - weighted variants allow flows to get different shares
Weighted Deficit Round-Robin

- Take turns sending
  - each queue may send up to $B$ bytes per turn
  - retain unused “credits” until next turn
  - $B$ should be at least max packet length
  - weighted version allows queue $i$ to send $B_i$ bytes per turn

- Characteristics of WDRR
  - simple, fast implementation – constant time per packet
  - makes traffic more bursty, if rates differ widely
    - e.g. 100 Mb/s link with 10 Mb/s flow, 900 at .1 Mb/s flows; fast flow sends bursts of 100, rather than one packet in 10
Packet Discard Policies

- Can influence fairness and TCP behavior
- Discard on arrival approach
  » discard arriving packet if queue above a threshold
  » or, set aside some buffer space as shared pool
    • e.g. each queue may use its “private space” plus a portion
      of shared area equal to 10% of the unused shared area
- Discard only when cannot avoid it
  » requires selecting a queue from which to discard
    • e.g. discard from longest queue
  » makes most efficient use of buffer space
  » more complex implementation
Discard from Longest Queue

- Uses data structure that orders queues by length
  - heap: $O(\log n)$ time per packet
  - can improve to $O(1)$ time since queue lengths change only by small amounts
  - cell-based version groups queues of same length together
  - packet version groups queues of about the same length
  - use “coarse grouping” to reduce number of separate lists
  - key is that adding or removing a packet does not move queue by more than one

- Diagram showing longest queues and queue numbers.
DRR with Discard from Longest Queue

- Smaller fluctuations than tail discard – still significant synchronization
“Sticky” Longest Queue Discard

- Queue State DRR continues to discard from same queue until it is the shortest non-empty queue.

low variation, even with small queues; low delay, no tuning
Time-Based Packet Scheduling

- Each queue has assigned bit rate
- After sending a packet, compute time for next packet using queue’s rate and packet length
  - e.g. if 10 Mb/s queue, sends a 500 bit packet, next packet can be sent in 50 µs
  - store next-send-times in a heap
- When earliest next-send-time arrives, send that queue’s first packet
  - use second heap to keep track of queues that recently became empty, so new arrivals won’t be sent too soon
- Resulting scheduler is not work-conserving
  - to make it w.c., advance time artificially (virtual time)
Hybrid Packet Schedulers

- Time-based schedulers require $O(\log(\# \text{ queues}))$ time per packet due to heap
- Hybrid strategies offer alternative approach
  - time-based scheduling for small number of rate classes
  - WDRR (or similar) scheduler for queues in same class
  - running time $O(\log(\# \text{ rate classes}))$
- Flows in same class have rates at most 2x apart
  - prevents creation of long bursts
  - 20 rate classes can cover rates from 10 Kb/s to 10 Gb/s
- Each rate class has dynamically varying rate equal to sum of rates of its active queues
Example Hybrid Scheduler

- Notation
  - \( q_i \): queue number \( i \)
  - \( r_i \): normalized rate of \( q_i \)
  - \( c_i \): rate class that includes \( q_i \)
  - \( R_j \): sum of rates of active flows in rate class \( j \)
  - \( D_j \): time to send next packet from class \( j \)

- Class \( j \) contains flows with rates in \((1/2^j, 2/2^j]\)

- Data structures
  - \textit{active}: heap of active classes
  - \textit{vactive}: heap of virtually active classes
  - \textit{wdrr}[j]: WRRR scheduling list for class \( j \)
Example Hybrid Scheduler

- On packet arrival for $q_i$ in class $j$
  - add packet to $q_i$
  - if $q_i$ was empty, add it to $wdrr[j]$, and increase $R_j$ by $r_i$
  - if $wdrr[j]$ was empty, add $j$ to active with $D_j = \text{now}$

- To select an outgoing packet
  - select class $j$ with smallest $D_j$ (using active)
  - if now<$D_j$, make now=$D_j$
  - use $wdrr[j]$ to select $q_i$ and send next packet from $q_i$
  - increase $D_j$ by (packet length)/$R_j$
  - if $q_i$ now empty, remove from $wdrr[j]$, reduce $R_j$ by $r_i$
  - if $wdrr[j]$ now empty, remove it from active and add it to vactive